CLAIMS

What is claimed is:

1. A method of expressing a target symbol sequence T relative to a reference symbol sequence R, the method comprising the steps of:

a) identifying a first longest common substring (LCS) of symbols in said sequences T and R;

b) defining said first LCS as a root node of a tree, said root node comprising said first LCS's starting position in said sequence R, either of said first LCS's length and said first LCS's ending position in said sequence R, and said first LCS's starting position in said sequence T, said root node being a parent node;

c) for each portion of said sequence T that precedes or succeeds said LCS in said sequence T:

d) where there is a portion of said sequence R corresponding to said portion of said sequence T:

e) identifying a subsequent longest common substring (LCS) of symbols in said portions;

f) if said subsequent LCS is identified, defining said subsequent LCS as a child node of said parent node, said child node comprising said subsequent LCS's starting position in said sequence R, either of said subsequent LCS's length and said subsequent LCS's ending position in said sequence R, and said subsequent LCS's starting position in said sequence T;

g) if said subsequent LCS is not identified, defining a child leaf of said parent node, said child leaf comprising the starting position of said portion of said sequence T in said sequence T and said portion of said sequence T itself; and

h) where there is no portion of said sequence R corresponding to said

portion of said sequence T, defining a child leaf of said parent node, said child leaf comprising the starting position of said portion of said sequence T in said sequence T and said portion of said sequence T itself.

2. A method according to claim 1 and further comprising recursively performing steps c) - h) for any LCS identified in any of said portions, thereby completely expressing said sequence T in said tree.

3. A method according to claim 1 and further comprising performing any of said steps a) - h) if said sequences R and T are alphanumeric text sequences

4. A method according to claim 1 and further comprising performing any of said steps a) - h) if said sequence T is a transformation of said sequence R.

5. A method according to claim 1 and further comprising performing any of said steps a) - h) if said sequence R is a word processing file which has undergone modifications to yield a modified word processing file as said sequence T.

6. A method according to claim 1 and further comprising storing any of said LCS nodes in a record comprising an identification byte of a predefined value indicating that said record is a node and a plurality of bytes for storing any of said LCS starting and ending positions and said LCS length.

7. A method according to claim 1 and further comprising storing any of said leaves in a record comprising an identification byte of a predefined value indicating that

said record is a leaf and a plurality of bytes for storing said starting position in said sequence and for storing said portion of said sequence T.

8. A method according to claim 1 and further comprising:

storing any of said LCS nodes in a record comprising an identification byte of a predefined value indicating that said record is a node and a plurality of bytes for storing any of said LCS starting and ending positions and said LCS length;

storing any of said leaves in a record comprising an identification byte of a predefined value indicating that said record is a leaf and a plurality of bytes for storing said starting position in said sequence and for storing said portion of said sequence T; and

storing any of said node and leaf records in a single data file having a header including the length of said sequence T followed by said node and leaf records in any order.

9. A method of reconstructing a target symbol sequence T having a known length using a reference symbol sequence R and a tree comprising any of:

at least one node, each node comprising the starting position of an LCS in said sequence R, either of said LCS's length and said LCS's ending position in said sequence R, and said LCS's starting position in said sequence T, and

at least one leaf, said leaf comprising the starting position of a portion of said sequence T in said sequence T and said portion of said sequence T itself, said tree completely expressing said sequence T,

the method comprising the steps of:

creating an array having a length equal to said length of said sequence T;

for each of said nodes:

retrieving an LCS within reference symbol sequence R at the starting position of said LCS in said sequence R indicated by said node and either of said LCS's length and said LCS's ending position in said sequence R indicated by said node; and

inserting said LCS into said array at said LCS's starting position in said sequence T indicated by the node; and

for each of said leaves:

inserting said portion of said sequence T stored within said leaf into said array at the position indicated by said leaf.

10. A method of expressing a target symbol sequence T relative to a reference symbol sequence R, the method comprising the steps of:

a) left-aligning said sequences T and R;

b) identifying a first longest common substring (LCS) of symbols in said sequences T and R starting at byte 0 of each sequence;

c) defining said first LCS as a root node of a tree, said root node comprising said first LCS's starting position in said sequence R, either of said first LCS's length and said first LCS's ending position in said sequence R, and said first LCS's starting position in said sequence T, said root node being a parent node;

d) for each portion of said sequence T that precedes said LCS in said sequence T:

e) where there is a portion of said sequence R corresponding to said preceding portion of said sequence T:

f) left-aligning said preceding and corresponding portions;

g) identifying a subsequent longest common substring (LCS) of symbols in said portions starting at byte 0 of each portion;

h) if said subsequent LCS is identified, defining said subsequent LCS as a child node of said parent node, said child node comprising said subsequent LCS's starting position in said sequence R, either of said subsequent LCS's length and said subsequent LCS's ending position in said sequence R, and said subsequent LCS's starting position in said sequence T;

i) if said subsequent LCS is not identified, defining a child leaf of said parent node, said child leaf comprising the starting position of said portion of said sequence T in said sequence T and said portion of said sequence T itself; and

j) where there is no portion of said sequence R corresponding to said portion of said sequence T, defining a child leaf of said parent node, said child leaf comprising the starting position of said portion of said sequence T in said sequence T and said portion of said sequence T itself; and

k) for each portion of said sequence T that succeeds said LCS in said sequence T:

l) where there is a portion of said sequence R corresponding to said succeeding portion of said sequence T:

m) right-aligning said succeeding and corresponding portions;

n) identifying a subsequent longest common substring (LCS) of symbols in said portions starting at the last byte of each portion;

o) if said subsequent LCS is identified, defining said subsequent LCS as a child node of said parent node, said child node comprising said subsequent LCS's starting position in said sequence R, either of said subsequent LCS's length and said subsequent LCS's ending position in said sequence R, and said subsequent LCS's starting position in said sequence T;

p) if said subsequent LCS is not identified, defining a child leaf

of said parent node, said child leaf comprising the starting position of said portion of said sequence T in said sequence T and said portion of said sequence T itself; and

q) where there is no portion of said sequence R corresponding to said portion of said sequence T, defining a child leaf of said parent node, said child leaf comprising the starting position of said portion of said sequence T in said sequence T and said portion of said sequence T itself.